QUIESCE TERMINATION/SUSPENSION
IN A MULTITHREADED ENVIRONMENT

Abstract

A method of coordinating the quiescing of the
5   additional threads of a multithreaded application to
facilitate the handling of an external event by a first
thread without interference from the additional threads. In
response to the detection of an external event by the first
thread, the first thread sends a quiesce event to each
10 additional thread of the application and suspends execution.
The quiesce event may be either a suspension event
requesting suspension of the additional threads or a
termination event requesting termination of the additional
threads. Each additional thread, upon receiving the quiesce
15 event, checks its environment to determine whether it is
holding any critical system resource. If the additional
thread determines that is not holding any critical system
resource and that it is therefore safe to quiesce, the
additional thread quiesces. Before quiescing, the last
20 additional thread to quiesce resumes the first thread, which
is now free to perform critical operations without
interference from the additional threads. If the quiesce
type is suspension, the first thread resumes the additional
threads upon completing its critical operations, whereupon
25 the application resumes its normal operation.